Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 34 (2014) 526 – 531

Procedia Computer Science

International Workshop on Communicating Objects and Machine to Machine for Mission-Critical Applications (COMMCA-2104)

Adaptive video streaming using TCP factors control with user parameters

Yassine Douga*[a], Malika Bourenane[a], Abdelhamid Mellouk[b].

[a]*University of Oran, Es Senia 31100, Algeria*
[b]*Paris-Est University,Crétei94010,Paris, France*

Abstract

Media streaming over TCP has become increasingly popular because TCP's congestion control provides remarkable stability to the Internet. Streaming over TCP requires adapting to bandwidth availability and other network parameters in order to have a good users satisfaction. Nowadays the streaming video can be projected on different kind of terminal as tablet, smart phones, laptop....etc. Each device has its own characteristics and parameters which must be taken into consideration on the video streaming adaptation process. In this paper, we propose an adaptive video streaming solution to improve the quality of experience (QoE) of the users by adapting TCP parameters to the user parameters on mobile networks. We validate the models using the ns2 simulator.

Selection and peer-review under responsibility of Conference Program Chairs

Keywords:QoE, TCP, Video streaming, Terminal device.

1. Introduction

Nowadays, the wide availability of wired and wireless broadband connections is enabling ubiquitous multimedia applications over the Internet, such as video streaming, personal video broadcasting, IPTV, and videoconferencing, at video resolutions that can scale up to full high definition (full HD, 1920x1080) at frame rates up to 30 fps. Such rich video contents require a compressed bit stream in the order of 10 Mbps along with adequate processing

* Corresponding author. Tel.: +213661234324;
E-mail address: maximusssse@hotmail.com

Selection and peer-review under responsibility of Conference Program Chairs
doi:10.1016/j.procs.2014.07.062

resources at the client for decoding. Nevertheless, the Internet is becoming more and more accessible to a wide spectrum of devices: if desktops users are normally equipped with large screens, good processing resources, and wired broadband connections, mobile users typically use small screen devices, with limited processing resources and wireless cellular connections that are characterized by variable link characteristics [2].

Thus, a key challenge is to provide the user with a seamless multimedia experience at the maximum Quality of Experience (QoE) that can be obtained given the available device and network resources. For this purpose, multimedia content must be made adaptive. It is important to notice that the adaptation process should take into account a wide set of variables such as user screen resolution, CPU load, network available bandwidth, power consumption, some of which are time-varying.Adaptive (live) video streaming represents a relevant advancement compared to classic progressive download streaming such as the one employed by YouTube [1].

In classic progressive download streaming, the video is delivered as any data using greedy TCP connections. The video stream is buffered at the receiver for a while before the playing is started so that short-term mismatches between the video bitrate and the available network bandwidth can be absorbed and video interruptions could be mitigated. Nevertheless, if the mismatch persists the buffer could eventually get empty and playback interruptions could occur affecting the user experience. On the other hand, with adaptive streaming the video source is adapted so that the user can watch videos at the maximum bitrate that is allowed by the time varying available bandwidth and by the device resources [1].

In this paper we focus on a particular adaptive streaming approach that is based on the user-satisfaction technique using TCP: the server encodes the video content at different bitrates and switches from one video version to another based on network feedbacks such as the measured available bandwidth. The variation of the value of these feedbacks is directly related to the variation of TCP factors, for example the measured available bandwidth has a direct relation with the TCP sliding window. In addition of adapting the transmitted video bitrates with the network feedbacks such as the measured available bandwidth, we adapt the TCP factors to the user parameters as terminal resolution, terminal CPU, power consumption..... By using this approach, we try to satisfy not only the network needs but also the user requirements.

The rest of the paper is organized as follows: Section 2 provides a brief review of the related works on adaptive streaming algorithms; we conclude this section by formulating the problem. Section 3 describes the proposed idea and its advantages. The simulation environment and the expected results are described in section 4. Finally, Section 6 concludes the paper.

2. Related works

In this part, we first present a review on the adaptive streaming techniques and then we focus on the most known commercial products providing adaptive streaming services.

2.1. Adaptive streaming techniques

In the last decade, a vast literature on video streaming has been produced. Main topics that have been investigated are: 1) the design of transport protocols specifically tailored for video streaming, 2) adaptation techniques, 3) scalable codecs.

Concerning the first topic, several transport protocols designed for video streaming have been proposed, such as the TCP Friendly Rate Control (TFRC), Real Time Streaming Protocol (RTSP), Microsoft Media Services (MMS) and Real Time Messaging Protocol (RTMP). Some of the mentioned protocols have been employed in commercial products such as Real Networks, Windows Media Player and Flash Player. Even though TCP has been regarded in the past as inappropriate for the transport of video streaming protocols, recently it is getting a wider acceptance and it is being used with the HTTP. This is mainly due to the following reasons: i) Internet applications are rapidly converging on web browsers; ii) HTTP-based streaming is cheaper to deploy since it employs standard HTTP servers; iii) TCP has built-in NAT traversal functionalities; iv) it is easy to be deployed within Content Delivery Networks (CDN)) TCP delivers most part of the Internet traffic and it is able to guarantee the stability of the network by means of an efficient congestion control algorithm.

In the Multimedia streaming via TCP approach, the authors develop analytic performance models to assess the performance of TCP when used to transport a live video streaming source without the use of quality adaptation. The theoretical results, obtained considering a constant bit rate (CBR) source and supported by an experimental evaluation, suggest that in order to achieve good performance in terms of startup delay and percentage of late packet arrivals, TCP requires a network bandwidth which is roughly two times the video bit rate. It is important to stress that such bandwidth over-provisioning would systematically waste half of the available bandwidth.

Regarding the adaptation techniques, different approaches have been proposed in the literature so far. The issue here is how to automatically throttle the video quality to match the available resources (network bandwidth, CPU) so that the user receives the video at the maximum possible quality. The proposed techniques to adapt the video source bitrate to the variable bandwidth can be classified into three main categories: 1) transcoding-based, 2) scalable encoding based, 3) stream-switching (or multiple-bitrate – MBR).

The transcoding-based approach (see Figure 1(a))), consists in adapting the video content to match a specific bitrate by means of on-the-fly transcoding of the raw content. These algorithms can achieve a very fine granularity by throttling frame rate, compression, and video resolution. Nevertheless, this comes at the cost of increased processing load and poor scalability, due to the fact that transcoding has to be done on a per-client basis. Another important issue is that such algorithms are difficult to be deployed in CDNs.

Another important class of adaptation algorithms (see Figure 1(a (b))) employs scalable codecs such as H264/MPEG-4 AVC. Both spatial and temporal scalability can be exploited to adapt picture resolution and frame rate without having to re-encode the raw video content. With respect to transcoding-based approach, scalable codecs reduce processing costs since the raw video is encoded once and adapted on-the-fly by exploiting the scalability features of the encoder. To be used with CDNs, this approach requires specialized servers implementing the adaptation logic. Also this approach is difficult to be used with CDNs since the adaptation logic requires to be run on specialized servers and content cannot be cached in standard proxies. Another issue is that the adaptation logic depends on the employed codec, thus restricting the content provider to use only a limited set of codecs.

Stream-switching algorithms (see Fig. 1(a (a)) encode the raw video content at increasing bitrates resulting into N versions, i.e. video levels; an algorithm dynamically chooses the video level that matches the user's available bandwidth; those algorithms minimize the processing costs since, once the video is encoded, no further processing is required in order to adapt the video to the variable bandwidth. Another important advantage of such algorithms is that they do not rely on particular functionalities of the employed codec and thus can be made codec-agnostic. The disadvantages of this approach are the increased storage requirements and the fact that adaptation is characterized by a coarser granularity since video bitrates can only belong to a discrete set of levels [4].

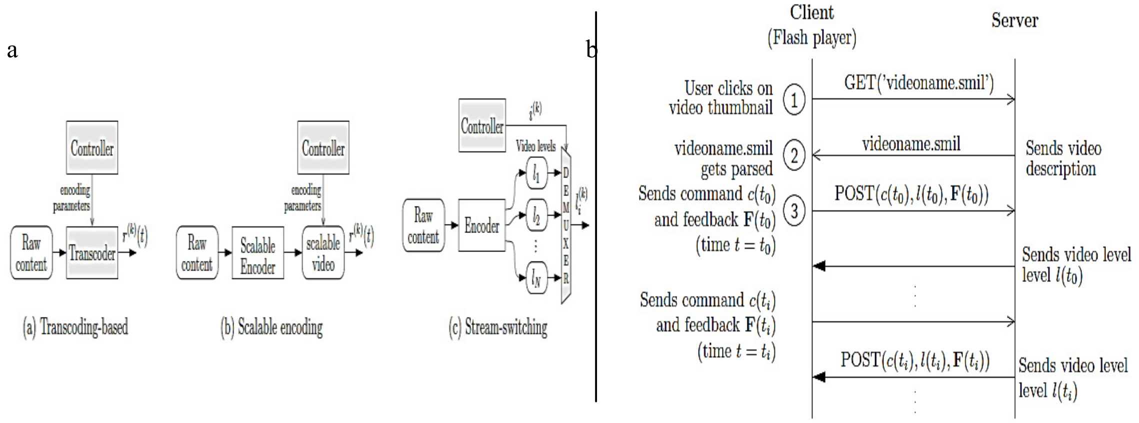

Fig. 1. (a) Adapting streaming techniques, (b) Client-server time sequence graph: thick lines represent video data transfer; thin lines represent HTTP requests sent from client to server.

3. Problem formulation

As we have said in the introduction, the streaming video service is one of the very used services on internet; one of the advantages of this service is that it can be used on several types of terminal, anywhere, any time and with any broadband internet technology. The classic streaming services stream the video from server to the user by taking into consideration just the state of the connection and available bandwidth, this will satisfy the network and data transmitting but not necessarily the user. Did the user receive the best possible video? What about the device, can it display the received video without problem?

The main goal of our work is to propose a solution that gives a maximum satisfaction for users according to available network parameters and user parameters such as the device parameters, the technology of user internet provider....

4. Proposed solution (user-based adaptive streaming (UAS))

As we have seen before, the used adaptive streaming techniques do not take into consideration the user satisfaction. In the proposed solution, we provide a new adaptive video streaming method. In this new method, in addition to the classic adaptive video streaming (CAS) we add a mechanism that will be executed before the classic adaptation and which adapts the network parameters according to the user factors. Thus, the selection of broadcasting video rates will depend on the network parameters adapted to those of the user. To tune the network parameters, we have chosen to adjust the factors of TCP. Such adjustment has a direct impact on the network parameters.

Thus, we have chosen the TCP sliding window to tune the available bandwidth.

- The TCP buffer.
- The TCP delays.
- The TCP jitter.

The user parameters that we have considered:

- Quality of source material of the terminal
- The terminal supported codecs.
- Resolution of the terminal
- Bit rate of the terminal
- Application layer video encoding of the terminal
- System and navigator CPU of the terminal
- Browser type.

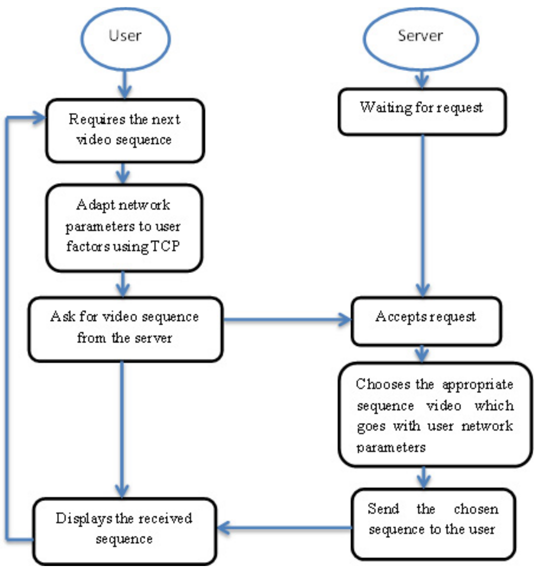

Fig.2. Datagram of the proposed idea.

5. Simulation and results

4.1. Simulation

We use the simulator NS2 to simulate our solution, because it is easy and adapted to this kind of simulation. First we create our streaming server which is connected to video data center; each video is saved with different bitrates. We also create our video streaming client; this client has many parameters to set as the bandwidth, the terminal resolution, the terminal CPU, the kind of the displaying device.... To evaluate the QoE we used Evalvid which us a video quality evaluation tool that can be used with NS2.

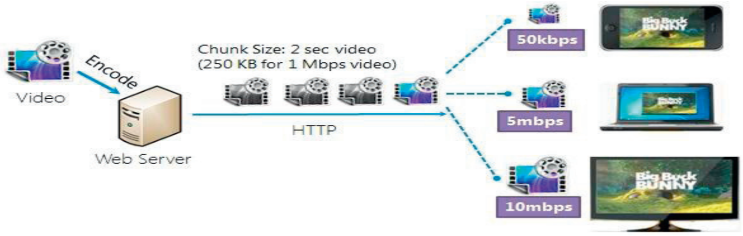

Fig.3. Simulation idea.

We have done two kinds of simulation:

- Simulation of classic adaptive streaming service:

We begin by doing this simulation to see how our model works and get results to compare it with those of the next simulation.

- Simulation of user adaptive streaming service:

In this simulation, we implement the proposed solution by adapting the streaming service to the user parameters. After doing each simulation, we measure the user satisfaction using a QoE tool, at each time, we compare the results of classic adaptive streaming service with those of user adaptive streaming service to see how far the proposed solution can satisfy the user comparing to the classic adaptive streaming. In addition of measuring the user satisfaction (QoE), we also measure the QoS of both of the simulated video streaming methods.

All simulation results are shown as a mark out of 10.On figure 4, we compare the user QoE results of the two studied methods according to the terminal resolution. Figure 5 compares the user QoE results of the two methods by taking into consideration the user bandwidth. Figure 6 shows the user QoE results of the two methods according to the terminal battery consumption.

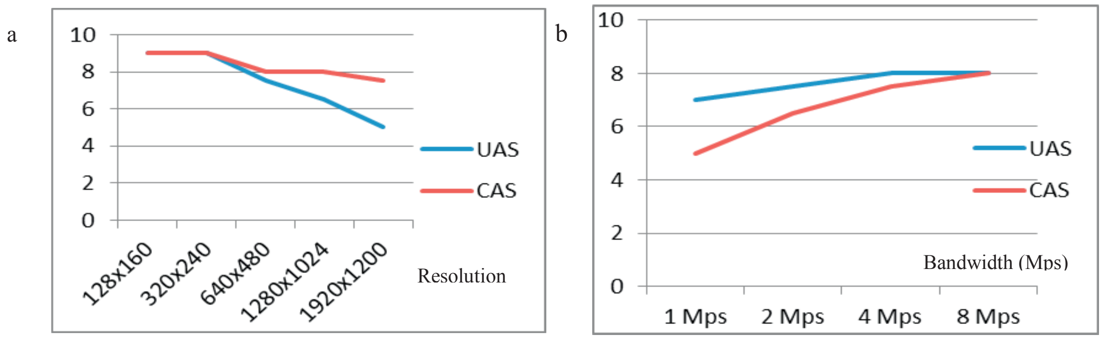

Fig.4. (a) QoE results according to the terminal resolution, (b) QoE results according to the bandwidth..

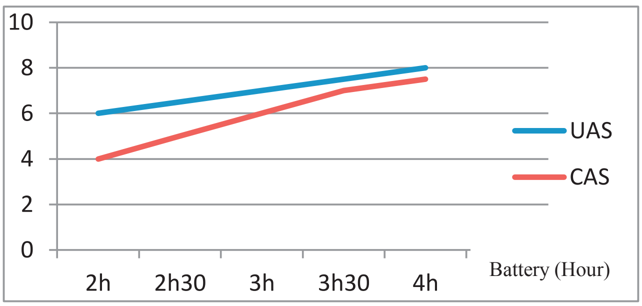

Fig.5.QoE results according to the battery consumption.

5.2 Simulation results discussion:

Simulation results show that the proposed solution 'UAS' gives better QoE results than the classic adaptive streaming solution 'CAS' by considering terminal resolution, for the lowest resolutions of the terminal we obtain the same results but by using higher terminal resolution values we obtain better results with ‘UAS’ than the ‘CAS’.

Also, for the available bandwidth of the network, the results show that with low value of available bandwidth of the network the ‘UAS’ solution adapts more better the video to the network condition than the ‘CAS’ and gives better QoE results.

About the battery consumption simulation part, we consider this as an inconvenient of our solution and we are working on it to find a solution because the ‘UAS’ consume more battery than ‘CAS’ because it does more operations to adapts the streaming flux than ‘CAS’ which increase the battery consumption.

6. Conclusion

In this paper, we proposed a new approach of adaptive streaming video; our main idea is to do the video adaptation according to the user parameters in addition to the classic adaptation methods. The simulation results have shown a good QoE performance and a better user satisfaction. To implement our idea, we used the transport protocol TCP as a bridge between the user and the network parameters. As a perspective, we aim to integrate additional user factors with the network factors to get better QoE and QoS results, and to provide a real implementation of our solution by developing a prototype.

7. Acknowledgement

The authors wish to express their sincere appreciation for financial support from the CELTIC+ European NOTTS 'Next generation over-the-top multimedia services' Project.

8. References

1. L. DeCicco, S. Mascolo and V. Palmisano, Feedback Control for Adaptive Live Video Streaming, ISBN: 978-1-4503-0518-1, Pages 145-156, ACM New York, NY, USA, 2011.
2. Yu, Y.-T and Tong, S.-R, ATCP+: An Adaptive TCP-Trunking Flow Control Scheme for Video Streaming, Wireless communications and mobile computing International Conference, Wireless communications and mobile computing; 1136-1140, ISBN: 9781615678648, New York, USA, 2009.
3. S. Linck, E. Mory, J. Bourgeois, E. Dedu and F. Spies, Adaptive multimedia streaming using a simulation test bed, Journal of Computational Science (2014), http://dx.doi.org/10.1016/j.jocs.2014.02.004.
4. IstvánKetykó, Katrien De Moor, Toon De Pessemier, Adrián Juan Verdejo, Kris Vanhecke, Wout Joseph, Luc Martens and Lieven De Marez, QoE Measurement of Mobile YouTube Video Streaming, MoViD '10 Proceedings of the 3rd workshop on Mobile video delivery, Pages 27-32, ISBN: 978-1-4503-0165-7, ACM New York, NY, USA, 2010.
5. C. Liu, I. Bouazizi, M. Gabbouj, Rate Adaptation for Adaptive HTTP Streaming, MMSys '11 Proceedings of the second annual ACM conference on Multimedia systems, Pages 169-174, ISBN: 978-1-4503-0518-1, ACM New York, NY, USA, 2011.
6. Kunwoo Park, Sangheon Pack and Ted, Taekyoung Kwon, An adaptive peer-to-peer live streaming system with incentives for resilience, computer Networks journal, Volume 54, Issue 8, Pages 1316–1327,2010.
7. K. Deep Singh, Y. Hadjadj-Aoul and G. Rubino, QUALITY OF EXPERIENCE ESTIMATION FOR ADAPTIVE HTTP/TCP VIDEO STREAMING USING H.264/AVC, CCNC - IEEE Consumer Communications & Networking Conference hal-00713723, version 1,2012.